# GitHub

1. **注册GitHub用户t**

<https://github.com/>

1. **新建仓库**

|  |
| --- |
| **一、登录GitHub**  **“+” --> “new repository”**  **“Repository name”中填写仓库名**  **其他根据需求选择**  **点击“Create repository”创建仓库** |
| **二、跳转后选择**  选择SSH协议（IOS系统可以使用HTTPS协议？） |
| snowc@SNOW-CITY MINGW64 /f/WorkSpace/geekbang/learn\_git (master)  $ **git remote add origin git@github.com:snowcitygame/learn\_git.git**  **三、# 给 本地仓库 添加 远程仓库（git remote add 远程库名 SSH字符串）**  snowc@SNOW-CITY MINGW64 /f/WorkSpace/geekbang/learn\_git (master)  $ **git remote**  origin  **# 测试是否和远程仓库连接，返回远程仓库名即为成功 （git remote）** |
| **四、# 配置SSH的公、私钥（ssh-keygen -t rsa -C “XXX@XXX”）**  snowc@SNOW-CITY MINGW64 /f/WorkSpace/geekbang/learn\_git (master)  $ **ssh-keygen -t rsa -C "snowcitygame@qq.com"**  Generating public/private rsa key pair.  Enter file in which to save the key (/c/Users/snowc/.ssh/id\_rsa):  Created directory '/c/Users/snowc/.ssh'.  Enter passphrase (empty for no passphrase):  Enter same passphrase again:  Your identification has been saved in /c/Users/snowc/.ssh/id\_rsa  Your public key has been saved in /c/Users/snowc/.ssh/id\_rsa.pub  The key fingerprint is:  SHA256:Z8o2DfJg+8eunGhK7hgHBUFJ0cyRfBlQQJ50Nz1S+Xo snowcitygame@qq.com  The key's randomart image is:  +---[RSA 3072]----+  | o\*X\*\*+ooo.. |  | .+\*oo...+ |  | +. . o |  | . . |  | . + S o. |  | . . \* \*. E |  | . o . \*... |  | \* .= oo |  | ..+o. =+. |  +----[SHA256]-----+  **五、# id\_rsa.pub为公钥 打开后复制内容**  **六、在GitHub中点击“头像”🡪“setting”🡪“SSH and GPG keys”🡪选择“New SSH key”**    **七、将公钥匙内容复制到Key中**    snowc@SNOW-CITY MINGW64 /f/WorkSpace/geekbang/learn\_git (master)  $ **ssh -T git@github.com**  Hi snowcitygame! You've successfully authenticated, but GitHub does not provide shell access.  **八、# 查看是否可以与远程仓库建立连接（ssh -T git@github.com）** |
| **九、新建好GitHub仓库后的使用**  …or create a new repository on the command line  echo "# test" >> README.md  # 本地仓库初始化  git init  # 本地仓库暂存  git add README.md  # 本地仓库提交  git commit -m "first commit"  # 给 本地仓库 添加 远程仓库（origin为远程仓库的别名，根据需求填写）  git remote add origin git@github.com:snowcitygame/test.git  #git remote add origin git@github.com:snowcitygame/learn\_git.git  # 将本地仓库数据推送到远程仓库（必须先配置好SSH的公、私钥）  git push -u origin master  …or push an existing repository from the command line  **# 给 本地仓库 添加 远程仓库**  **git remote add origin git@github.com:snowcitygame/test.git**  **# 将本地仓库数据推送到远程仓库**  **git push -u origin master**  …or import code from another repository  You can initialize this repository with code from a Subversion, Mercurial, or TFS project. |
| git add [<options>] [--] <pathspec>...  -n, --dry-run dry run  -v, --verbose be verbose  -i, --interactive interactive picking  -p, --patch select hunks interactively  -e, --edit edit current diff and apply  -f, --force allow adding otherwise ignored files  -u, --update update tracked files  --renormalize renormalize EOL of tracked files (implies -u)  -N, --intent-to-add record only the fact that the path will be added later  -A, --all add changes from all tracked and untracked files  --ignore-removal ignore paths removed in the working tree (same as --no-all)  --refresh don't add, only refresh the index  --ignore-errors just skip files which cannot be added because of errors  --ignore-missing check if - even missing - files are ignored in dry run  --chmod (+|-)x override the executable bit of the listed files  --pathspec-from-file <file> |
| git commit [<options>] [--] <pathspec>...  -q, --quiet suppress summary after successful commit  -v, --verbose show diff in commit message template  Commit message options  -F, --file <file> read message from file  --author <author> override author for commit  --date <date> override date for commit  -m, --message <message>  commit message  -c, --reedit-message <commit>  reuse and edit message from specified commit  -C, --reuse-message <commit>  reuse message from specified commit  --fixup <commit> use autosquash formatted message to fixup specified commit  --squash <commit> use autosquash formatted message to squash specified commit  --reset-author the commit is authored by me now (used with -C/-c/--amend)  -s, --signoff add Signed-off-by:  -t, --template <file>  use specified template file  -e, --edit force edit of commit  --cleanup <mode> how to strip spaces and #comments from message  --status include status in commit message template  -S, --gpg-sign[=<key-id>]  GPG sign commit  Commit contents options  -a, --all commit all changed files  -i, --include add specified files to index for commit  --interactive interactively add files  -p, --patch interactively add changes  -o, --only commit only specified files  -n, --no-verify bypass pre-commit and commit-msg hooks  --dry-run show what would be committed  --short show status concisely  --branch show branch information  --ahead-behind compute full ahead/behind values  --porcelain machine-readable output  --long show status in long format (default)  -z, --null terminate entries with NUL  --amend amend previous commit  --no-post-rewrite bypass post-rewrite hook  -u, --untracked-files[=<mode>]  show untracked files, optional modes: all, normal, no. (Default: all)  --pathspec-from-file <file>  read pathspec from file  --pathspec-file-nul with --pathspec-from-file, pathspec elements are separated with NUL character |
| git remote [-v | --verbose]  git remote add [-t <branch>] [-m <master>] [-f] [--tags | --no-tags] [--mirror=<fetch|push>] <name> <url>  git remote rename <old> <new>  git remote remove <name>  git remote set-head <name> (-a | --auto | -d | --delete | <branch>)  git remote [-v | --verbose] show [-n] <name>  git remote prune [-n | --dry-run] <name>  git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]  git remote set-branches [--add] <name> <branch>...  git remote get-url [--push] [--all] <name>  git remote set-url [--push] <name> <newurl> [<oldurl>]  git remote set-url --add <name> <newurl>  git remote set-url --delete <name> <url> |
| git push [<options>] [<repository> [<refspec>...]]  -v, --verbose be more verbose  -q, --quiet be more quiet  --repo <repository> repository  --all push all refs  --mirror mirror all refs  -d, --delete delete refs  --tags push tags (can't be used with --all or --mirror)  -n, --dry-run dry run  --porcelain machine-readable output  -f, --force force updates  --force-with-lease[=<refname>:<expect>]  require old value of ref to be at this value  --recurse-submodules (check|on-demand|no)  control recursive pushing of submodules  --thin use thin pack  --receive-pack <receive-pack>  receive pack program  --exec <receive-pack>  receive pack program  -u, --set-upstream set upstream for git pull/status  --progress force progress reporting  --prune prune locally removed refs  --no-verify bypass pre-push hook  --follow-tags push missing but relevant tags  --signed[=(yes|no|if-asked)]  GPG sign the push  --atomic request atomic transaction on remote side  -o, --push-option <server-specific>  option to transmit  -4, --ipv4 use IPv4 addresses only  -6, --ipv6 use IPv6 addresses only |

1. **Fork其他GitHub仓库、Clone到本地**

|  |
| --- |
| 1.进入原GitHub仓库 --> 点击‘Fork’  原GitHub仓库内容会完整复制到自己的GitHub仓库中  2.进入fork后的自己的GitHub仓库  3.点击‘Clone’-->选择SSH协议-->复制该仓库地址  4.在本地使用git clone命令，从远程GitHub仓库克隆到本地仓库  使用git clone的目录不能已是仓库  git clone Clone按钮中SSH协议的仓库地址  git clone git@github.com:Python001-class01/Python001-class01.git  以上步骤结束后，在指定目录下会下载有GitHub仓库对应的文件 |
| 5.在本地进行文件操  git add  git commit -m 'xxx'  push -u xxx master |